

Programming with Alice

Program Design

Why Design First?

Good program design involves thinking about the problem **before writing any code**.

The design phase allows the steps of the solution to be identified and refined without the restrictions of the programming language.

The design process is generally **iterative**, meaning the design can be reconsidered and modified multiple times.

Example Problem (in Alice) Animating A Flying Bird

As a first pass, we might think of the solution in terms of only a few general statements.

bird flaps wings

bird moves forward

Animating A Flying Bird - v.2

Unless your Bird class in Alice comes with a flap method, you will need more detail. What is involved in flapping wings?

both wings move up
both wings move down
bird moves forward

Animating A Flying Bird - v.3

At this point, we are getting conceptually close to a reasonable solution, but there is now the issue of timing. The design process helps us identify which steps are simultaneous, and which occur in order.

left wing moves up
right wing moves up
left wing moves down
right wing moves down
bird moves forward

Animating A Flying Bird - v.4

left wing moves up
right wing moves up

at the same time

left wing moves down
right wing moves down

at the same time

bird moves forward

Animating A Flying Bird - v.5

left wing moves up
right wing moves up

at the same time

left wing moves down
right wing moves down

at the same time

left wing to neutral
right wing to neutral

at the same time

bird moves forward

over entire time

Flying Bird – Alice Code

world.my first method ()

No variables

Do together

Do in order

Do together

pterodactyl2.rightWing roll left 1/8 revolutions

pterodactyl2.leftWing roll right 0.12 revolutions

Do together

pterodactyl2.rightWing roll right 0.25 revolutions

pterodactyl2.leftWing roll left 0.25 revolutions

Do together

pterodactyl2.rightWing roll left 1/8 revolutions

pterodactyl2.leftWing roll right 0.12 revolutions

pterodactyl2 move forward 1 meter duration = 3 seconds

Pseudocode

Any structured collection of commands that approximate programming instructions is called pseudocode.

This code cannot be executed on a computer, but it can be translated into a programming language.

Although it is often necessary to consider your target language (e.g., Alice, Java, Turing), the same pseudocode should be applicable to all languages (with some adjustments).